

به نام خدا



گزارش کار تمرین الگوریتم های ژنتیک

استفاده از الگوریتم ژنتیک برای پیدا کردن نقطه کمینه تابع تست الگوریتم های بهینه سازی – Schaffer N.4

استاد راهنما

سرکار خانم فاطمه زریسفی کرمانی

گروه

سامیه بنی اسدی

سید امیرحسین ادهمی میرحسینی

دانشگاه شهید باهنر کرمان – علوم کامپیوتر

پاییز ۱۳۹۷

مقدمه

الگوریتم ژنتیک یک الگوریتم فراابتکاری الهام گرفته شده از روند انتخاب طبیعی است که متعلق به کلاس بزرگتر الگوریتم‌های تکاملی (فرگشتی) است. الگوریتم‌های ژنتیک بطور معمول برای بدست آوردن جواب‌های بهینه برای مسائل جست‌وجو و بهینه‌سازی براساس بقای برترین‌ها یا انتخاب طبیعی و تکنیک‌های الهام گرفته شده از زیست‌شناسی مانند همبری (Crossover)، جهش (Mutation) و انتخاب (Selection) است. الگوریتم‌های ژنتیک ابزار سودمندی در بازشناسی-الگو، انتخاب‌ویژگی، درک تصویر و یادگیری ماشین است.

جان هالند (John Holland) در سال ۱۹۶۰ الگوریتم ژنتیک را بر اساس مدل تکاملی داروین معرفی کرد؛ بعد از آن، شاگرد او گلدبرگ (Goldberg) در سال ۱۹۸۹ الگوریتم ژنتیک را توسعه داد.

در مدل‌سازی الگوریتم ژنتیک یک تکنیک برنامه‌نویسی است که از تکامل ژنتیکی به عنوان یک الگوی حل مسئله استفاده می‌کند. مسئله‌ای که باید حل شود دارای ورودی‌هایی می‌باشد که طی یک فرایند الگوبرداری شده از تکامل ژنتیکی به راه‌حلها تبدیل می‌شود سپس راه حلها به عنوان کاندیدها توسط تابع ارزیاب (Fitness Function) مورد ارزیابی قرار می‌گیرند و چنانچه شرط خروج مسئله فراهم شده باشد الگوریتم خاتمه می‌یابد. بطور کلی یک الگوریتم مبتنی بر تکرار است که اغلب بخش‌های آن به صورت فرایندهای تصادفی انتخاب می‌شوند که این الگوریتم‌ها از بخش‌های تابع برازش، انتخاب، همبری و جهش تشکیل می‌شوند.

هدف

هدف ما استفاده از الگوریتم ژنتیک برای پیدا نکردن گلوبال تابع Schaffer N.4 است.

طبق اطلاعات ویکی‌پدیا این تابع در نقطه (0,1.25313) گلوبال مینیمم با مقدار 0.292579 دارد. [\[۱\]](#)

نرم افزار و دانش مورد نیاز

نرم افزار: MATLAB

دانش: آشنایی ابتدایی با الگوریتم ژنتیک - آشنایی ابتدایی با برنامه‌نویسی و مفاهیم اولیه آن - آشنایی ابتدایی با برنامه‌نویسی در MATLAB

شرح کار

ابتدا پارامترهای ژنتیک را تعریف می‌کنیم به این صورت که طول کروموزوم ۲ (تعداد ژن‌ها) بدلیل اینکه دو متغیر x و y به عنوان ورودی تابع هستند. بازه مقادیر x و y هر دو از -۱۰ تا ۱۰ می‌باشد.

جمعیت اولیه ای به صورت اعداد تصادفی در بازه تعریف شده برای x و y را در ابتدا تولید می‌کنیم. سپس جمعیت را به تابع شایستگی داده (Fitness Function) و شایستگی جمعیت را حساب می‌کنیم تابع شایستگی به صورت زیر است:

$$fitness\ function = \frac{1}{0.5 + \frac{\cos^2[\sin(|x^2 - y^2|)] - 0.5}{[1 + 0.001(x^2 + y^2)]^2}}$$

دلیل اینکه در اینجا تابع را معکوس کردیم این است که در واقع تابع مینیمم دارد (با مشتق گرفتن از تابع و برابر صفر قرار دادن آن و از طرف دیگر شکل تابع) اما ما به دنبال این هستیم که تابع شایستگی مقدار ماکزیمم را پیدا کند پس آن را معکوس کرده.

حال که شایستگی جمعیت را حساب کردیم احتمال انتخاب را برای اعضا محاسبه می کنیم تا در چرخ رولت از آنها استفاده کنیم.

محاسبه احتمال انتخاب به روش زیر انجام می شود:

$$selection\ probability = \frac{fit_i}{\sum_{j=1}^N fit_j}$$

پس از محاسبه احتمال انتخاب برای انتخاب بهترین جواب ماکزیمم مقدار شایستگی ها را پیدا کرده و کروموزومی که بیشترین شایستگی را دارد به عنوان جواب در این تکرار انتخاب می کنیم در ضمن پس از هر بار تکرار الگوریتم بهترین شایستگی آن تکرار را ذخیره کرده.

سپس استفاده از احتمال انتخابی که در مرحله قبل بدست آوردیم چرخ رولت را می سازیم. دلیل استفاده از چرخ رولت این است که به اعضای برتر جمعیت (نقاطی که به نقطه مینیمم مدنظر ما نزدیک تر هستند) شانس بیشتری برای انتخاب بدهیم و برای آن نقاطی که مناسب نیستند شانس کمتر.

روند کار پس از محاسبه احتمال انتخاب و قرار دادن آنها در یک وکتور به صورت تجمعی (در واقع انگار چرخ را به صورت نوار(وکتور) کشیدیم) نوبت به ساخت حوضچه جفت گیری می رسد برای انکار به این صورت عمل می کنیم که برای انتخاب اعضای حوضچه ازدواج یک عدد تصادفی در نظر میگیریم بین صفر و یک، و اولین عضوی که احتمال تجمعی اش بیشتر یا برابر عدد تصادفی بود، انتخاب می گردد. توجه داشته باشید باید به تعداد اعضا این رویه تکرار شود، و در این رویه به تعداد اعضا باید احتمال تجمعی با عدد تصادفی مقایسه شوند و در صورت انتخاب، دیگر عدد تصادفی با بقیه اعضا مقایسه نشود.

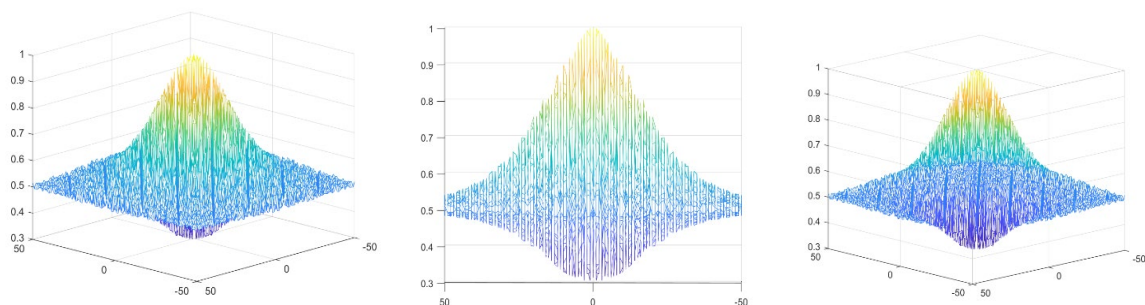
پس از ساختن حوضچه ازدواج نوبت به همبری (crossover) اعضای که در حوضچه ازدواج هستند می رسد، نوع عملگر انتخاب شده همبری تک نقطه ای (Single Point Crossover) است.

پس از اعمال عملگر همبری جمعیت جدید را با جمعیت قدیم جایگزین کرده و عملگر جهش را که از نوع گوسی است روی آن اعمال می کنیم.

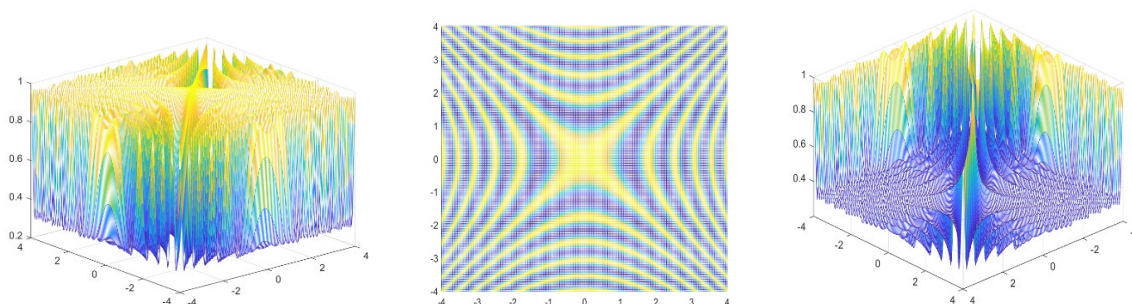
این روند تا رسیدن به شرط توقف ادامه می یابد که در اینجا شرط توسط رسیدن به نزدیکی نقطه مورد نظر یا رسیدن به تعداد تکرار خاصی است.

نتایج

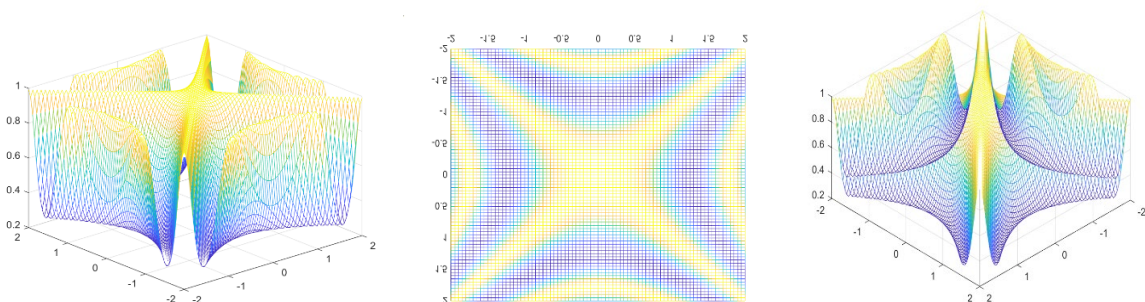
شکل نمودار کلی در بازه‌های مختلف به صورت زیر می‌باشد:



عکس ۱) شکل تابع در بازه $[-50, 50:1]$ به ترتیب از چپ: بالا - دو بعدی - پایین



عکس ۲) شکل تابع در بازه $[-4, 4:0.05]$ به ترتیب از چپ: بالا - دو بعدی - پایین



عکس ۳) شکل تابع در بازه $[-2, 2:0.05]$ به ترتیب از چپ: بالا - دو بعدی - پایین

نتایج زیر پس از تغییر پارامترهای اندازه جمعیت (N)، نرخ همبری (Pc)، نرخ جهش (Pm)، ضریب انحراف معیار تابع گوسی (scale) و تعداد تکرار بدست آمده.

N	Pc	Pm	Scale	maximum Iterations	Mean Iterations	Successful Runs
100	0.5	0.1	0.1	300	254	24/100
100	0.5	0.1	0.1	500	366	46/100
100	0.8	0.1	0.1	500	362	44/100
100	0.8	0.5	0.1	500	454	19/100
150	0.8	0.1	0.1	500	291	60/100
150	1	0.1	0.1	500	269	68/100
150	1	0.05	0.1	500	136	93/100
150	1	0.04	0.1	500	160	89/100
150	1	0.03	0.1	500	171	89/100
150	1	0.045	0.1	500	164	93/100
150	1	0.035	0.1	500	169	84/100
150	1	0.045	0.2	500	135	97/100
150	1	0.045	0.3	500	109	99/100
150	1	0.045	0.4	500	108	99/100
150	1	0.045	0.5	500	94	99/100
150	1	0.045	0.6	400	93	99/100
150	1	0.045	0.8	500	98	100/100
150	1	0.045	0.8	300	101	97/100

در واقع از ابتدا معلوم است که الگوریتم سریعاً به جواب نزدیک می‌شود پس نرخ همبری باید بالا باشد از طرفی نرخ جهش باید پایین باشد تا جوابی که بدست آمده زیاد تغییر نکند و روی همان جواب تاکید کند، جهش کم فقط این موضوع رو تضمین می‌کند که الگوریتم در بهینه محلی (خط‌هایی که در شکل‌های ۲ و ۳ هم از بالا هم از پایین معلوم و به شکل یک علامت جمع هستند) گیر نکند.